

## Prova P1

### Instruções:

- Prova sem consulta e individual.
- É permitido o uso de calculadora científica.
- A prova é composta por 5 questões, todas com o peso de 2 pontos.
- Responda as questões justificando os seus resultados e apresentado a memória de cálculo.
- Responda as questões na folha pautada de respostas. Respostas nesta folha não serão consideradas.

1. (2 pontos) Defina a abordagem greedy e comente uma das formas de garantir que os algoritmos greedy sempre encontram uma das soluções ótimas?
2. (2 pontos) Defina a abordagem de divisão e conquista e de um exemplo de algoritmo que utiliza essa abordagem.
3. (2 pontos) Resolva as seguintes recorrências utilizando o método mestre:

- a.  $T(n) = 4T\left(\frac{n}{16}\right) + 1$
- b.  $T(n) = 16T\left(\frac{n}{32}\right) + n^{\frac{1}{3}}$
- c.  $T(n) = 4T\left(\frac{n}{16}\right) + n$
- d.  $T(n) = 9T\left(\frac{n}{5}\right) + n^3$

#### TEOREMA T.1 O método mestre

- ❖ Caso 1: Se  $f(n) = O(n^{\log_b a - \epsilon})$  para alguma constante  $\epsilon > 0$ , então  $T(n) = \Theta(n^{\log_b a})$ .
- ❖ Caso 2: Se  $f(n) = \Theta(n^{\log_b a})$ , então  $T(n) = \Theta(n^{\log_b a} \log_2 n)$ .
- ❖ Caso 3: Se  $f(n) = \Omega(n^{\log_b a + \epsilon})$  para alguma constante  $\epsilon > 0$ , e se  $af(n/b) \leq cf(n)$  para alguma constante  $c < 1$  e todos os  $n$  suficientemente grandes, então  $T(n) = \Theta(f(n))$ .

4. (2 pontos) Com base no algoritmo de Huffman codes responda as questões abaixo:

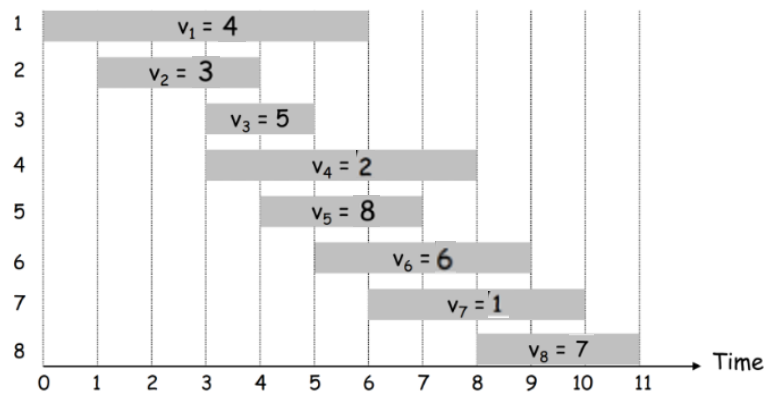
```

HUFFMAN(S)
1  n = |S|
2  Q = S
3  for i = 1 to n - 1
4      allocate a new node w
5      y = EXTRACT-MIN(Q)
6      z = EXTRACT-MIN(Q)
7      w.left = z
8      w.right = y
9      w.freq = z.freq + y.freq
10     INSERT(Q, w)
11 return EXTRACT-MIN(Q) // the root of the tree is the only node left
    
```

- a. O Huffman codes utiliza que abordagem algorítmica para compressão de dados?
- b. Qual o tempo de execução do algoritmo?
- c. Defina a árvore final para o seguinte dicionário (letra:frequência): {[a:12], [b:11], [c:8], [d:17], [e:24], [f:5]}.
- d. Descreva os códigos definidos pela árvore do item c.

5. (2 pontos) Utilize a abordagem de programação dinâmica com memoização para resolver a alocação de tarefas ponderadas abaixo.

$$OPT(j) = \begin{cases} 0 & , se j = 0 \\ \max(v_j + OPT(p(j)), OPT(j-1)) & , se j > 0 \end{cases}$$



Responda as seguintes perguntas:

- O que é memoização?
- Calcule  $p(j)$  para todas as requisições.
- Calcule  $OPT(8)$ .
- Indique quais requisições fazem parte da solução ótima dada por  $OPT(8)$ .